RIGA TECHNICAL UNIVERSITY

Mārtiņš Mihaeljans

EVALUATION OF ADAPTIVE SOLUTIONS FOR MANAGEMENT OF VIRTUALIZED SOFTWARE-DEFINED NETWORKS

Summary of the Doctoral Thesis

RTU Press
Riga 2025

RIGA TECHNICAL UNIVERSITY
Faculty of Computer Science, Information Technology and Energy
Institute of Photonics, Electronics and Telecommunications

Mārtiņš Mihaeljans
Doctoral Student of the Study Programme "Telecommunications"

EVALUATION OF ADAPTIVE SOLUTIONS FOR MANAGEMENT OF VIRTUALIZED SOFTWARE-DEFINED NETWORKS

Summary of the Doctoral Thesis

Scientific supervisor
Professor Dr. sc. ing.
JURĢIS PORIŅŠ

Scientific consultant
Dr. sc. ing.
ANDRIS SKRASTIŅŠ

RTU Press
Riga 2025

Mihaeljans, M. Evaluation of Adaptive Solutions for Management of Virtualized Software-Defined Networks. Summary of the Doctoral Thesis. Riga: RTU Press, 2025. 36 p.

Published in accordance with the decision of the Promotion Council "P-08" of 27 June 2025, Minutes No. 40.

Cover picture from www.shutterstock.com.

https://doi.org/10.7250/9789934372216
ISBN 978-9934-37-221-6 (pdf)

DOCTORAL THESIS PROPOSED TO RIGA TECHNICAL UNIVERSITY FOR PROMOTION TO THE SCIENTIFIC DEGREE OF DOCTOR OF SCIENCE

To be granted the scientific degree of Doctor of Science (Ph.D), the present Doctoral Thesis has been submitted for defense at the open meeting of RTU Promotion Council on 14 November 2025 at 11.00 at the Faculty of Computer Science, Information Technology and Energy of Riga Technical University, Āzenes iela 12, Room 201.

OFFICIAL REVIEWERS

Associate Professor Dr. sc. ing. Andis Supe,
Riga Technical University

Senior Researcher Dr. sc. comp. Atis Elsts,
Institute of Electronics and Computer Science, Latvia

Associate Professor Ph.D Oskars Java
Vidzeme University of Applied Sciences, Latvia

DECLARATION OF ACADEMIC INTEGRITY

I hereby declare that the Doctoral Thesis submitted for review to Riga Technical University for promotion to the scientific degree of Doctor of Science (Ph.D) is my own. I confirm that this Doctoral Thesis has not been submitted to any other university for promotion to a scientific degree.

Mārtiņš Mihaeljans ……. (signature)
Date: ……

The Doctoral Thesis has been written in Latvian. It consists of an Introduction, 8 chapters, Conclusions, 40 figures, and 8 appendices; the total number of pages is 134, including appendices. The Bibliography contains 68 titles.

ABSTRACT

This PhD Thesis evaluates adaptive solutions for management of virtualized software-defined networks, thus making a step forward in the advancement of the information world. This research has originated from the exploration of general principles of conventional networking, software-defined networking (SDN) and its capabilities, the Internet of Things (IoT) with its constrained network device properties, and state-of-the-art intent-based networking (IBN).

The Ph.D Thesis is a collection of eight research results that not only contribute to the provision of seven new adaptive solutions for telecommunication technologies but also, at the time of writing, serves as insight into novel and actual technologies in the industry itself. Problems discussed are: network path generation, fragmentation mitigation in the IoT networks, cut on usage of specific-purpose built network equipment, and automation of human-managed network configuration.

Research results covered in this Thesis have been published in scientific publications, which are attached as appendices to the full-length text. The adaptive solutions proposed are: a service function path encapsulation used when necessary, a reactive service path discovery, service function share, service function path declaration, recommendations in specification for IoT things, recommendations for Thread mesh network density compliance, and recommendations for lift of IBN adaptation.

Hypotheses proposed in this Thesis are all based on published research. Hypotheses have been proven to be true, as adaptive solutions do solve problems discussed in the problem statement. For compliance with existing global network infrastructure, the adaptive solutions may require modification according to the targeted working principle.

ACKNOWLEDGMENTS

Most importantly, I want to thank my family, who provided immense support, thus allowing me to focus on conducting this research and on the studies themself. It is all 10 years spent in the university that counts here.

I thank my mentor, Andris Skrastiņš. He not only took part in all aspects of conjuring the science, but also had a significant impact and set a good example on how to approach and execute work in general.

I also to thank my trainer, Alberts Bagojans. To be a part of the RTU team of athletes has been a positive experience throughout many years.

It is my supervisor Jurģis Poriņš and his RTU team who made these Doctoral Studies possible for me in the first place. I am grateful for that and the support that the University has provided.

CONTENTS

Introduction 8
Background and topicality of the study 9
Problem statement 9
Objective and tasks of the study 10
Hypotheses 10
Author's contributions 11
Scientific novelty 11
Practical significance 12
Publications and approbation of the Thesis 13
Overview of thesis sections 14
Section 1. Introduction of technologies 14
Section 2. Network topology-aware SF chaining 19
Section 3. Reactive SF path detection 21
Section 4. SF share in inter-data center networks 23
Section 5. SR-MPLS on re-programmable data plane 24
Section 6. Analysis of the Internet of Things 26
Section 7. Thread mesh density considerations 27
Section 8. Intent-based networking analysis 29
Conclusions 31
References 35

ABBREVIATIONS

The abbreviations used in the Thesis Summary.

AI – Artificial Intelligence
CLI – Command Line Interface
CPU – Central Processing Unit
DC – Data Center
DLP – Data Leak Prevention
DPI – Deep Packet Inspection
FW – Firewall
GAN – Generative Adversarial Network
GUI – Graphical User Interface
IBN – Intent-Based Networking
IMT-2020 – International Mobile Telecommunications-2020 standard
IoT – the Internet of Things
IP – Internet Protocol
IPS – Intrusion Prevention System
ISO – International Organization of Standardization
ITU-T – International Telecommunication Union Telecommunication Standardization Sector
LAN – Local Area Network
LB – Load Balancer
ML – Machine Learning
MPLS – Multiprotocol Label Switching
NAT – Network Address Translation
NF – Network Function
NFV – Network Function Virtualization
NLP – Natural Language Processing
NSH – Network Service Header
OSI – Open System Interconnection
PAN – Personal Area Network
PBNM – Policy-Based Network Management
PISA – Protocol-Independent Switch Architecture
PSA – Programmable Switch Architecture
P4 – Programmable Protocol-independent Packet Processing
RAM – Random Access Memory
SDN – Software Defined Networking
SF – Service Function
SFF – Service Function Forwarder
SFC – Service Function Chaining
SID – Segment Identifier
SR-MPLS – Segment Routing with MPLS data plane
TE – Threat Emulation/Extraction
VLAN – Virtual Local Area Network

INTRODUCTION

The General Assembly of the United Nations has a resolution of sustainable development goals where it lists man-made resource management and distribution mechanisms – power grid, water supply, food distribution, sanitary, military, monetary, and healthcare system – which are all locked in an information world's global network of physical and logical nodes. Its interconnected systems' operation and development maintains the well-being of all to the network attached ecosystems and among their residents [1].

Since the dawn of digitalization, a cardinal component of the information world is packet-switched networking. Communication, enabled by the network, allows its users to exchange information, goods, and services [2]. Evaluation of adaptive solutions for management of virtualized software-defined networks (in this Ph.D Thesis) is based on multiple research studies in packet-switched networks. These studies result in the development and evaluation of seven new adaptive solutions.

Multihoming (a single target node is simultaneously reachable through multiple connections) gets enabled for service function chaining. This functionality allows for the usage of service function sharing for synchronous processing of multipath data flows.

An alternative to the usage of full encapsulation for the length of the service function path is introduced. This alternative method is the use of encapsulation only in the segments of the network where it is necessary.

Predefined service function path policy is replaced with a mechanism for the automation of path discovery via the use of default path and network traffic re-classification. This functionality mitigates human intervention in network management, therefore, eliminating the possibility of a network disruption, which is a result of misconfiguration.

A recommendation is developed for combating missing things' categorization in the Internet of Things. It allows grouping devices according to their physical attributes. Categorization enables one sort of things to have a joint attribute policy (reachability, identifiability, software support, energy consumption type), thus increasing the compatibility among them.

An analysis is conducted for Thread mesh density (Thread is a close-distance communication technology for IoT devices in small-sized personal area networks). The analysis results in a recommendation that allows easing the network availability issue caused by personal area link regeneration due to changes in network topology.

A programmable packet processing solution model is developed for service function chaining over segment routing with MPLS transport. This model identifies the relation between the segment identifier and MPLS label, and proposes a service path description with the use of unified identification.

A working principles modification is proposed for an artificial intelligence and machine learning run state-of-the-art intent-based networking. Modification simplifies tasks that are delegated to the artificial intelligence by splitting management types.

This Thesis has four major parts. First, an explanation of covered technologies is given. Second, the research done is outlined with illustrations tailored to the context of this Thesis. Third, conclusions about the results of research are drawn. Fourth, background research is provided in the Thesis appendices . The research and its results discussed in this Thesis have been published in eight scientific articles (all indexed in Scopus) to improve the operation of telecommunications and computer network infrastructure.

Background and topicality of the study

This research has originated from the exploration of general principles of conventional networking, software-defined networking (SDN) and its capabilities of service function chaining, the Internet of Things (IoT) with its constrained network device properties, and state-of-the-art intent-based networking. All mentioned technologies are pertinent to the communication and information technology field.

As the research base is a collection of work carried out from the fall of 2020 till the spring of 2025, it not only showcases the rapid development and innovation in the field itself, but also delivers solutions that are in synchronization with the telecommunication industry.

And even more so, it surpasses the commercial off-the-shelf product offer as research showcases technology implementation that is not yet widely available – programmable switch architecture (PSA). It is a technology intended for data packet processing customization according to individual requirements. On PSA, a service function chaining domain had been constructed and evaluated.

Problem statement

It is typical for the information world, also known as the digital medium, to be unpredictable and in constant change. The changes tend to result in various network problems. Some examples of these problems are: remotely located network node link disjoint due to unavailable network segments, live stream connection errors due to uneven dispersion of content load, voice signal jams due to low network quality. All mentioned problems are directly addressable as their solution resides in the analysis of network nodes participating in the communication. Meanwhile, not all problems are directly addressable. Problems for which solutions can be achieved in various ways, and those that are not directly related to the misbehavior of a specific network entity, are discussed in this work:

1. Enhancement of network path generation and coordination of transmitted information in them by using service function chaining.
2. Fragmentation mitigation of the Internet of Things by using unified communication standards to increase compatibility among devices in use.
3. Decrease the use of task-specific or proprietary devices by advancing the feature set for re-programmable units.
4. Automation of human-managed network configuration, maintenance, and telemetry analysis by providing intent-based networking.

The mentioned problems are in the scope of the technology list given in the study's background and explored in Section 1. Acknowledging the existing problems, the research tasks and hypotheses are defined.

Objective and tasks of the study

The research objective is an analysis of existing solutions and the development of new adaptive solutions set for software-defined networking and complementary technologies in compliance with future network and IoT frameworks, including IMT-2020 non-radio aspects, defined in ITU-T Y recommendations. The research scope is packet-switched networks in data centers, edge, access networks, and the Internet of Things. Solutions are made in a way to ensure that they are compatible with existing technologies and incorporate recommendations or standardization of the working principles of the mentioned technologies. Synchronization is also evident through the fact that the references of the Doctoral Thesis hold multiple recommendations and technology standards.

Defined tasks for the achievement of the set objective are as follows:

1. Perform analysis of software-defined networking and related technologies to define the problem statement.
2. Perform experiments and emulate working principles of software-defined networks and complementary technologies and the Internet of Things to give recommendations about proposed and evaluated solutions for the elimination of given problems.
3. From the output of analysis and experiments define a solution set in accordance with the intended telecommunication framework.
4. Evaluate the scenarios of enhancement possibilities of the developed solution according to technologies' evolvement trends, such as a spike in device count, a move towards packet forwarding, or a push for virtualization and automation.

The tasks define guidelines for actions performable on all in this work discussed problems. **The scientific method used in six of eight studies is the conduct of an experiment by network topology emulation. In two other studies, the scientific method used is a literature analysis**.

Hypotheses

Seven solutions are produced in this Thesis to answer the following hypotheses:

1. Reactive service function path discovery and coordination of transmitted information along multiple administrative domains is achievable via the use of partial-encapsulation and re-classification methods in service function chaining.
2. Fragmentation of the Internet of Things can be mitigated with the use of a unified transmission medium, and device compatibility can rise with the use of a unified application layer, if more specific characteristics of things are defined.
3. Service function chaining via segment routing can be built on top of programmable devices, thus serving as an alternative to the use of purpose-specific or proprietary devices with similar capabilities.
4. Computer network creation and management can be automated by the use of intent-based networking if artificial intelligence functionality is delegated separately to different network management types.

The first hypothesis holds in multiple studies, as service function chaining is the most explored technology in this work. This technology is mostly used in data center networks.

The third hypothesis also deals with service function chaining (SFC), but in a different way. Here, SFC is a subject, and the object is re-programmable devices. This refers to experiments with programmable switch architecture.

Author's contributions

The Ph.D Thesis is a collection of the results of eight studies that not only contribute to the provision of new adaptive solutions for telecommunication technologies, but also, at the time of writing, serve as insight into novel and actual technologies in the industry itself.

The most studied technology is service function chaining (SFC). In five of eight studies, adaptive solutions are proposed for overcoming limitations of SFC technology. These solutions do not require their architecture modification as they are defined in a way of reimagining the use of existing elements.

The Thesis also touches on well-studied but not well-rounded technology of the Internet of Things. A retrospective of both standardization and conducted studies revealed a causal relation between inadequately general recommendations and interoperability issues.

The strive for focus on the coverage of industry's actual research topics reveals itself most prominently through the more recent studies of programmable protocol-independent packet processing and intent-based networking technologies.

Scientific novelty

As a result of conducted research an applicable to software defined networking and related technologies, and the internet of things novelties have been produced:

- Two new service function chaining packet encapsulation methods [3].
- New service function chaining path policy creation approach [4], [5].
- New categorization concept for physical devices of the internet of things to overcome the existing device interoperability issue [6].
- New use case for devices within the service function domain for service of multipath data flows [7].
- New causal relation between mesh networks link count and networks regeneration occurrence ratio in Thread technology [8].
- New service function path declaration approach in segment routing [9], and programs for network devices of programmable switch architecture [10].
- Two new intent-based networking models that are aimed at easing tasks meant for artificial intelligence, with models customized in accordance with a manageable network [11].

Practical significance

The Ph.D thesis does not have a single definitive problem or a single hypothesis to be proven. There are multiple of them. Also, the tasks put in place are written to give guidance on how the study cases were approached. The background of the research are stand-alone scientific publications. Therefore, the research outcomes addressed within the context of this Ph.D Thesis, as adaptive solutions, are as follows:

1. A service function path encapsulation according to the necessity for service function chaining, in opposition to encapsulation throughout.
2. A reactive service path discovery with the use of a default path and data flow reclassification, in opposition to proactive (defined before the arrival of the ingress traffic classification event) manual path policy.
3. Service function share in inter-datacenter networks that enables the ability to expand the service function path's set of available network functions, or enables redundancy for existing network functions.
4. Service function path declaration with designated segment identifier in segment routing built on top of reprogrammable devices.
5. Recommendation on the implementation of specific characteristics for IoT things definition.
6. Recommendation on IoT Thread mesh network density (count of connections) compliance, to avoid regeneration of the whole network by defining the minimal required number of connections that must form between a routing-capable device and the attached end-device.
7. Recommendation for modification of intent-based networking that envisions ease of tasks meant for artificial intelligence according to a manageable network.

All seven adaptive solutions are a valuable contribution to the information world as they are thought out and constructed in compliance with existing technology in use. Especially the service function chaining rooted solutions, which do not require modification in SFC domains standards.

Solutions that are expressed as recommendations carry their significance within the conducted studies context itself. For example, the IoT study of the 5th solution holds multiple paragraphs of arguments pointing out the flaws in existing standardization.

All aspects of the introduction section are revisited in detail in later sections and overall summarized in the conclusions section.

Publications and approbation of the Thesis

Research results covered in this Thesis have been published in scientific publications that are attached as appendices and listed below. Conferences where results were presented are also listed.

Scientific publications

1. M. Mihaeljans and A. Skrastins, "Network Topology-aware Service Function Chaining in Software Defined Network," 2020 28th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2020, pp. 1–4, doi: 10.1109/TELFOR51502.2020.9306554.
2. M. Mihaeljans and A. Skrastins, "Reactive Service Function Path Discovery Approach in Software Defined Network," 2021 29th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2021, pp. 1–4, doi: 10.1109/TELFOR52709.2021.9653356.
3. M. Mihaeljans and A. Skrastins, "Evaluation of reactive service function path discovery in symmetrical environment," Telfor Journal, vol. 14, no. 1, pp. 2–7, 2022, doi: 10.5937/telfor2201002M.
4. M. Mihaeljans and A. Skrastins, "IoT concept and SDN fusion in consumer products: Overview," 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Tenerife, Canary Islands, Spain, 2023, pp. 1–6, doi: 10.1109/ICECCME57830.2023.10252518.
5. M. Mihaeljans and A. Skrastins, "Openthread Network Density Evaluation: Quantitative Analysis," 2023 Symposium on Internet of Things (SIoT), São Paulo, Brazil, 2023, pp. 1–5, doi: 10.1109/SIoT60039.2023.10390236.
6. M. Mihaeljans and A. Skrastins, "Efficient Multipath Service Function Chaining in Inter-Data Center Networks," 2023 31st Telecommunications Forum (TELFOR), Belgrade, Serbia, 2023, pp. 1–4, doi: 10.1109/TELFOR59449.2023.10372615.
7. M. Mihaeljans, A. Skrastins, and J. Porins, "Paramounts of Intent-Based Networking," 28th International Conference ELECTRONICS 2024, Palanga, Lithuania, September 2024.
8. M. Mihaeljans, A. Skrastins, and J. Porins, "Service Function Chaining via SR-MPLS over P4: A rudimentary analysis," The International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS2024), Dubrovnik, Croatia, September 2024.

Result presentations in conferences

1. 28th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2020.
2. 29th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2021.
3. 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Tenerife, Canary Islands, Spain, 2023.
4. Symposium on Internet of Things (SIoT), São Paulo, Brazil, 2023.
5. 31st Telecommunications Forum (TELFOR), Belgrade, Serbia, 2023.
6. 28th International Conference ELECTRONICS 2024, Palanga, Lithuania, September 2024.
7. The International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS2024), Dubrovnik, Croatia, September 2024.

OVERVIEW OF THE THESIS SECTIONS

Section 1. Introduction of technologies

Software-defined network stands in contrast to conventional networking. In legacy devices, all tasks (transmission, control, management) were performed in each device separately. Whereas a software-defined network (SDN) consists of three planes, each representing another task: data forwarding, control, and management, as shown in Fig. 1.

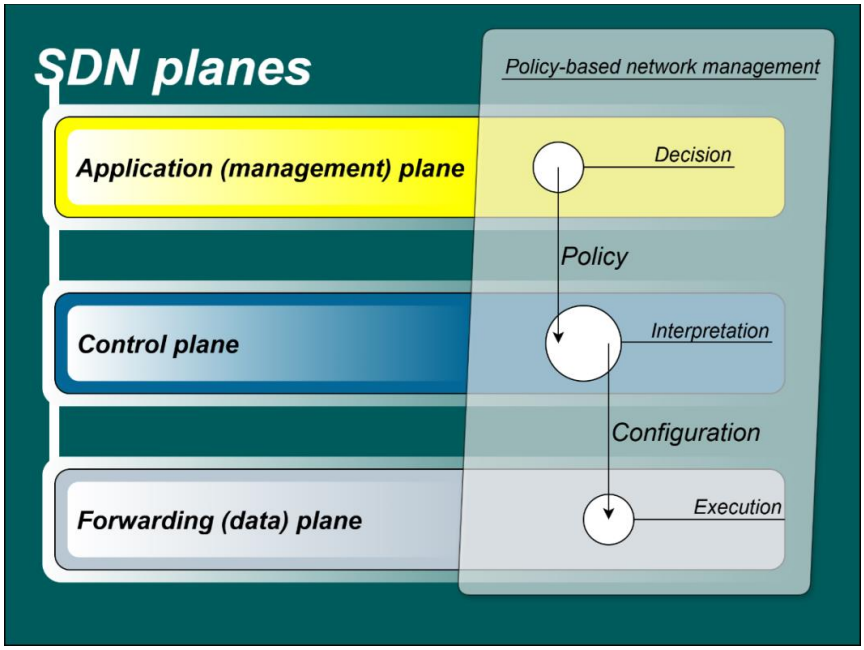

Fig. 1. Planes of a software-defined network.

Policy-based network management (PBNM) is a management paradigm that separates the rules governing the behavior of a system from its functionality. It promises to reduce maintenance costs of information systems while improving flexibility and adaptability [12].

Application plane provides human interaction with the network via a graphical user interface (GUI) or a command line interface (CLI). This plane consists of applications facilitating the interaction. The interaction itself is a set of tasks that are man-imposed and machine-committed. The control plane's goal is to provide centralized control of all forwarding plane elements and a single point of view of network state. The forwarding plane mostly consists of all the same kind of network equipment devices that perform data flow forwarding. In legacy networks, data packets are switched and routed according to the open systems interconnection model of the International Organization for Standardization (ISO OSI). In SDN, data packets are forwarded according to the forwarding policy defined in the match-action pipeline. Figure 2 shows SDN, where the control function is separated from the forwarding and security network equipment and provided using a remote controller through a control channel.

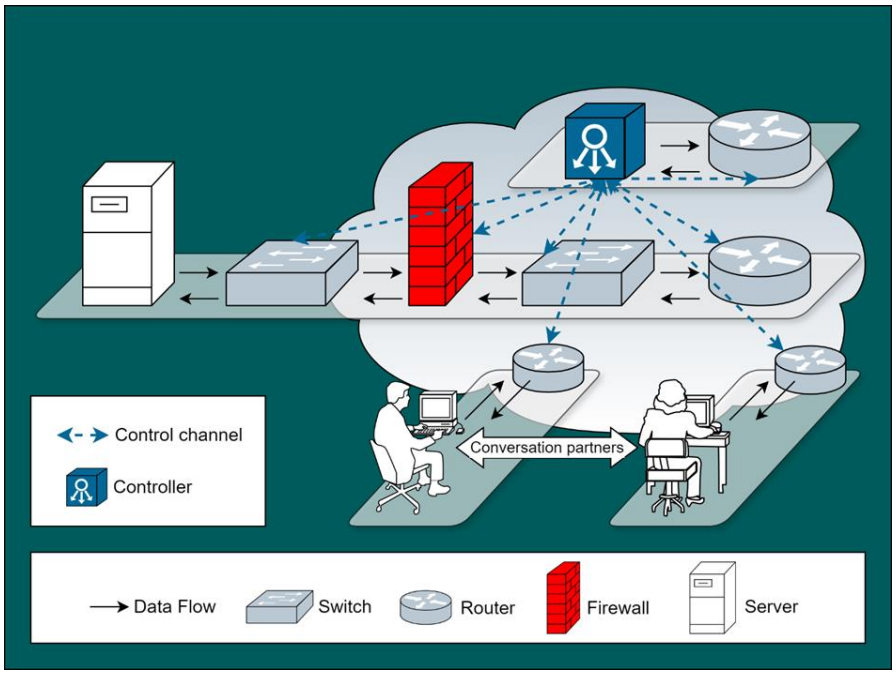

Fig. 2. Software-defined network.

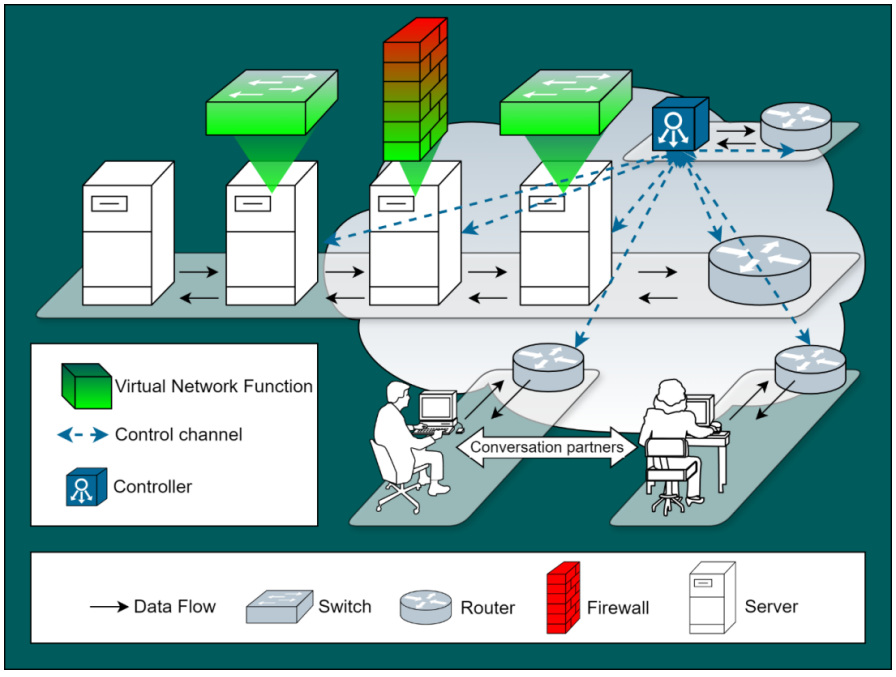

Fig 3. Network function virtualization.

Network function virtualization (NFV) is where specific-purpose built network equipment is replaced by general-purpose devices on which the network functions (NFs) are emulated. Network functions are all actions performed upon transferred data except the transference itself (switching, routing, forwarding). NFV is shown in Fig. 3.

NFs are: firewall (FW), network address translation (NAT), load balancing (LB), deep packet inspection (DPI), proxy, antispam, antivirus, intrusion prevention system (IPS), data leak prevention (DLP), and threat emulation/extraction (TE).

A turning point in data center (DC) maturity was the introduction of general-purpose hardware. It is a hardware designed to accommodate virtualized services on top of a shared central processing unit (CPU), random access memory (RAM), storage, and other resources.

Service function chaining (SFC) is a data flow steering mechanism through a specific service function path with the use of data packet encapsulation. It allows for changing data packet processing without making a change in transport topology. Packet encapsulation is an approach where a header is added on top of the original data packet, usually above or below the ISO OSI 2nd layer header, depending on the transmission medium. In SFC terminology, service function equals network function. SFC is shown in Fig. 4.

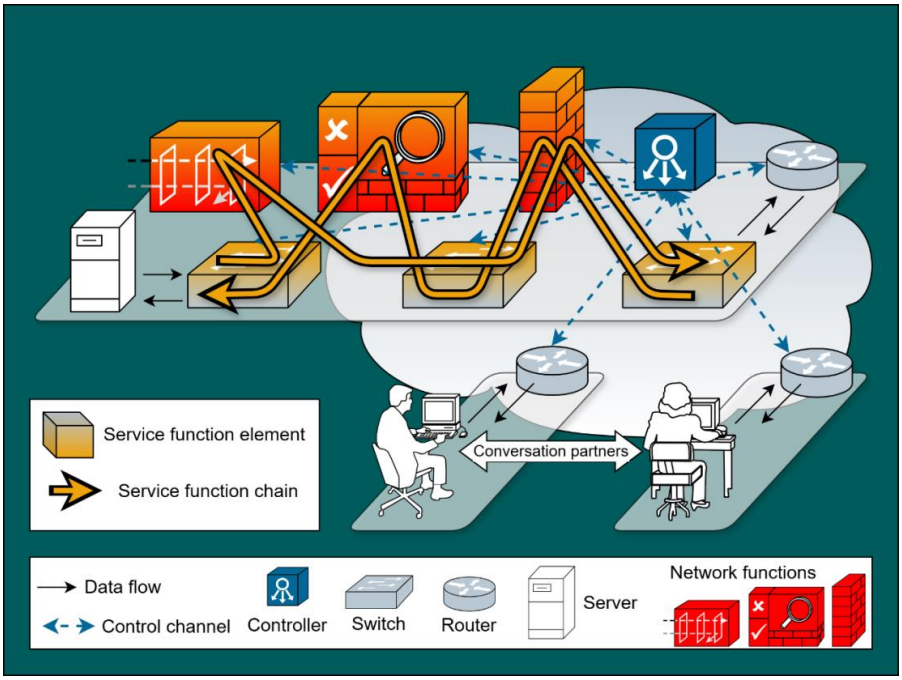

Fig. 4. Service function chaining.

The Internet of Things is a network that links everyday objects to a digital medium. In Fig. 5, a device of the Internet of Things (IoT) is represented as a black box, as it can be anything – from home appliances to industrial or environmental objects. Interchangeable terms for the physical-world and information-world within the context of IoT are real-world and digital-world. Figure 5 also shows how the IoT devices can communicate with one another and their placement in the network.

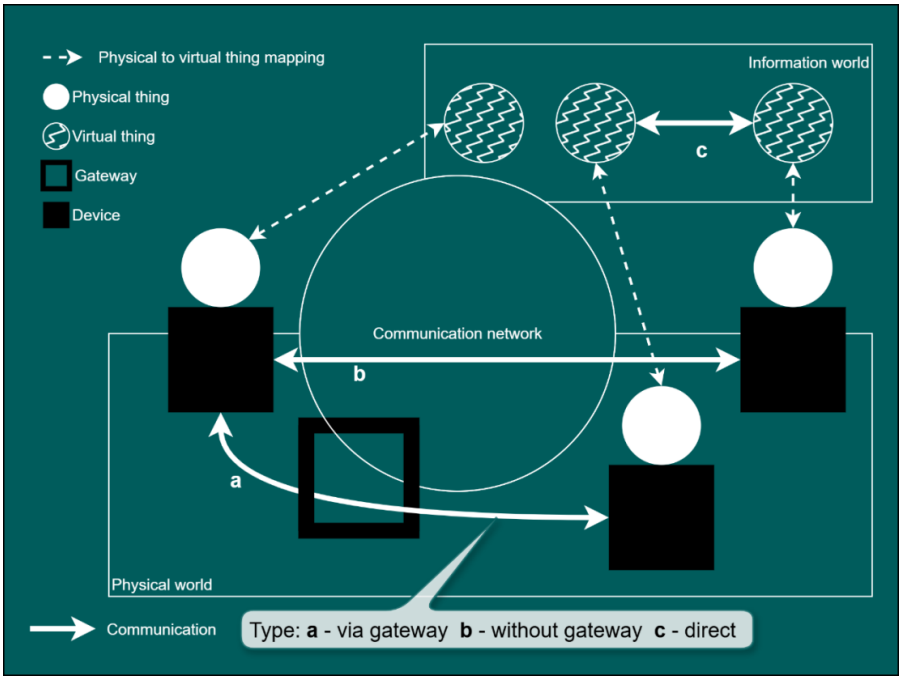

Fig. 5. The Internet of Things.

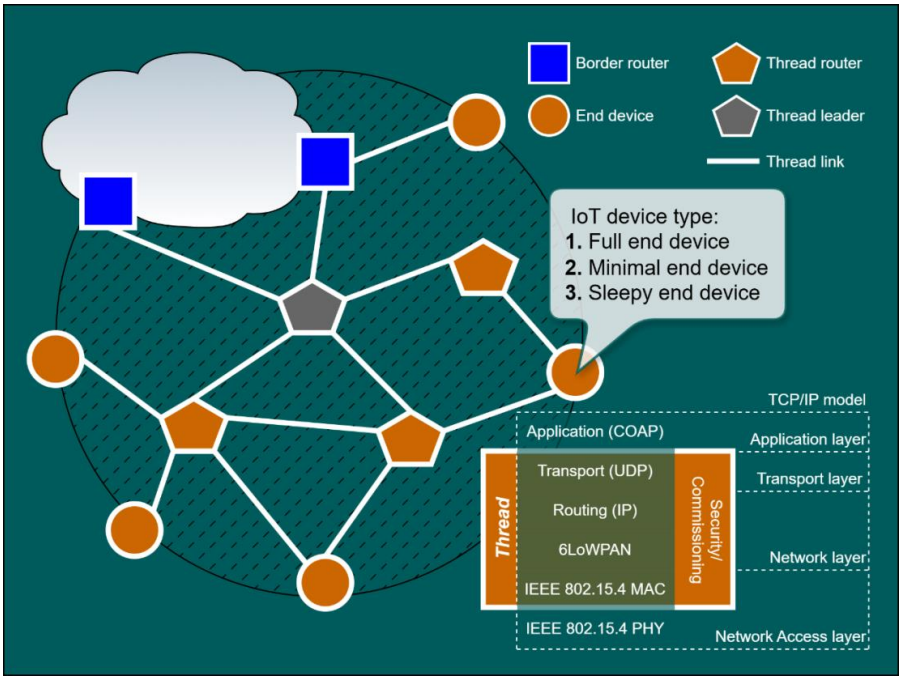

Fig. 6. Thread mesh network.

Thread mesh network and Matter protocol enable IoT things interconnectivity locally. IoT things, that are built as consumer products, face three big problems: incompatibility among them in the physical data layer, incompatibility among them in the application layer, and the allocation of their control element in the cloud. Device manufacturers identified these problems and made a work group that developed a solution – Thread technology and Matter protocol.

Matter uses a common application layer and data model that delivers interoperability between devices, allowing them to communicate with each other across multiple IP network technologies. Since its launch, Matter runs on Wi-Fi, Thread, and Ethernet network layers and uses Bluetooth Low Energy for commissioning [13].

Matter treats networks as shared resources. It makes no stipulation of exclusive network ownership or access. As a result, it is possible to overlay multiple Matter networks over the same set of constituent IP networks [14]. Therefore, device functions can be managed from multiple sources simultaneously.

Personal area network (PAN) created with the use of Thread and Thread's placement in TCP/IP model is shown in Fig. 6. In this figure Thread network is connected to the cloud via the use of a border router. Even though Thread is an IP-based communication technology, it is meant to be used in local networks.

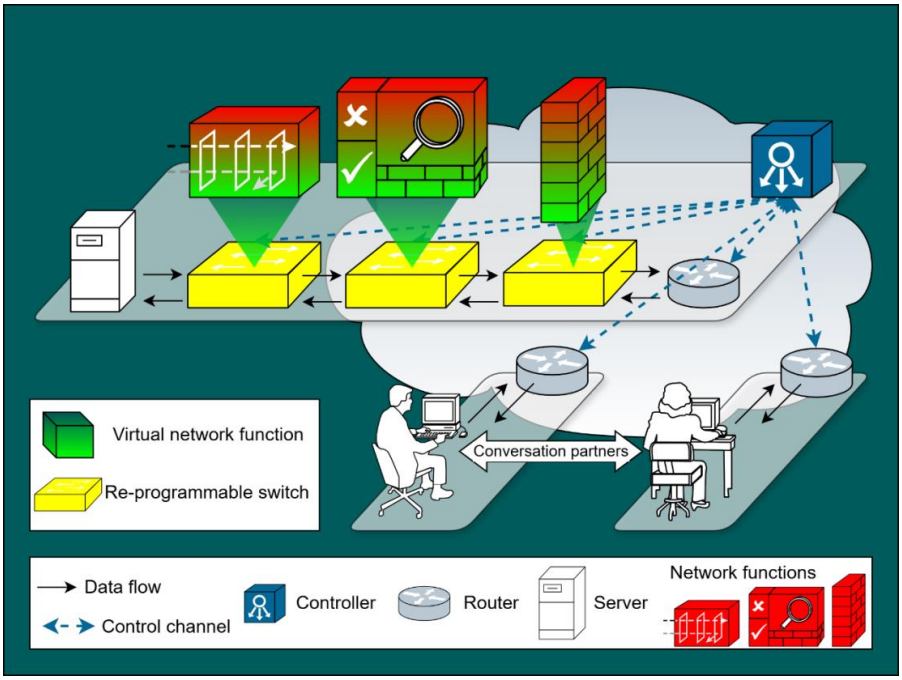

Fig. 7. Programmable protocol-independent packet processing.

Programmable protocol-independent packet processing (P4) is a concept of enabling re-programmability on network equipment devices. P4 language is used on devices whose architecture corresponds to that of a protocol-independent switch architecture (PISA). In SDN, the popular OpenFlow protocol supporting architecture was made to program the control plane, but the match-action pipeline stayed dependent on network protocols. P4, on the other hand, is made to program the data plane [15].

Shown in Fig. 7 is P4, where network functions are virtualized and run on re-programmable switches. These switches are controlled remotely. The development of network functions on PISA architecture is a challenge in telecommunications engineering, and at the same time, it is a necessity that boosts SDN technologies' maturity.

Segment routing has regained interest in the industry thanks to the fact that, on top of it, the service function chaining can be done with MPLS or IPv6 as transport. Thus, it removes barriers among legacy and SDN devices.

Segment routing with MPLS (SR-MPLS) is a packet routing method that combines source routing with multiprotocol label switching (MPLS) encapsulation. In segment routing, the source host or the first router in the path pushes an encapsulation on the original packet. This encapsulation is used by transit routers to make decisions on how to route the packet. Segment routing enables service providers to support the realization of network slicing in an IP/MPLS transport network. The network as a whole, in a distributed and entirely automated manner, can share a single infrastructure resource along multiple virtual service slices. For example, one network slice is optimized continuously for low-cost transport, a second one is optimized continuously for low-latency transport, and a third one is orchestrated to support disjoint services, etc. [16].

Intent-based networking (IBN) enables its users to make requests, aka intents, out of which a network policy can be generated. The user only contributes to the matter of what the required goal is to achieve, leaving how to do it to artificial intelligence (AI), for example:

- segment the branch office from headquarters to ensure data leak prevention;
- limit any excessive use of available bandwidth for media streaming.

IBN is a network management technology that is made to mitigate human intervention in network creation and oversight by the use of artificial intelligence and machine learning (ML). Starting from intent profiling with natural language processing (NLP) to network availability foresight via big data and generative adversarial networks (GAN) [17]–[21].

To affect policies in the network, entities within the network must interpret prescribed policies. Not all entities within a network necessarily possess the ability to interpret policies directly. Such entities may require assistance in interpreting policies [22]. Thus, the structure of IBN must be clearly defined and must allow network policy interpretation into configuration.

Section 2. Network topology-aware SF chaining

Network function virtualization (NFV) applications and service providers' legacy network equipment must be capable of coexisting and be compatible with existing network management and orchestration. NFV needs to ensure migration from proprietary physical appliances to virtual on open standards-based software-defined solutions. In other words, NFV must be capable of working in hybrid mode between legacy physical and virtual network nodes [23].

To answer the need for orchestration in a changing environment, network topology-aware service function chaining was developed and evaluated in this work. Its core value is to remove encapsulation everywhere where it is not necessary. A modification of the original architecture of service function chaining (SFC) is not required. The difference between full and partial SF encapsulation is shown in Fig. 8, where a partially encapsulated SF path leaves the SF that is used by both communication directions with no encapsulation.

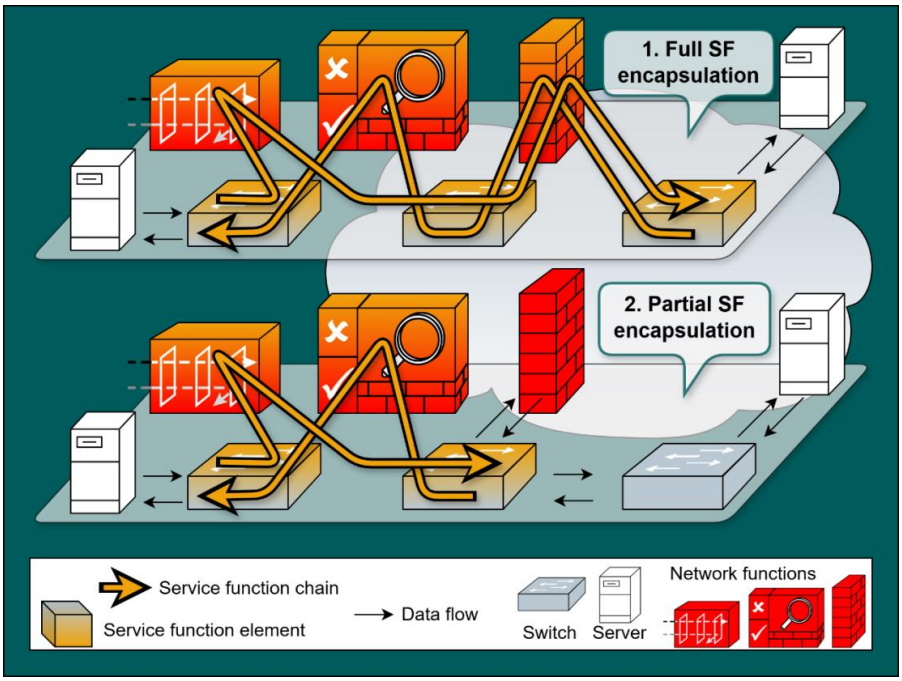

Fig. 8. Partial SF encapsulation.

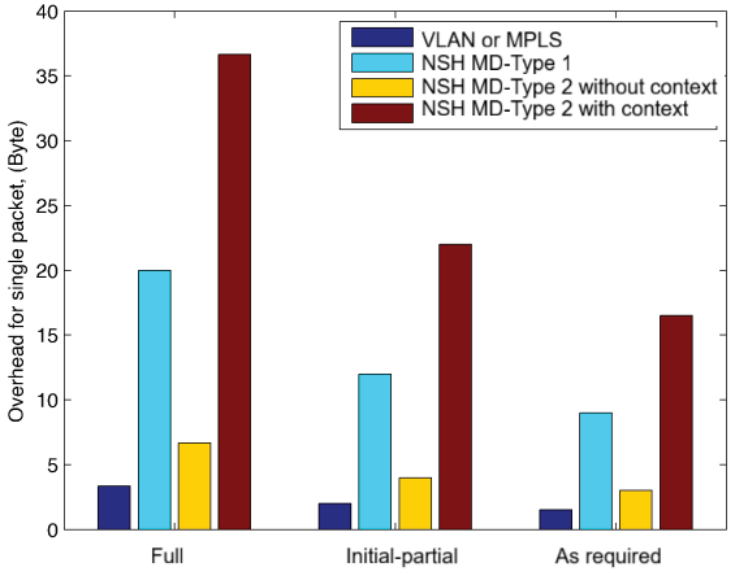

Fig. 9. Comparison of encapsulation application approaches.

In Fig. 9, the result of service function domain emulation is shown. The graph has a direct dependence on emulated network topology (two-tier star topology with three service functions). In other cases, service function count and topology type can differ,

and depend on network size or other factors. This fact does not remove the validity of the results that clearly indicate the ability to decrease the overhead of data packet transmission. Discussed overhead refers to an additional header applied to data packets when they cross a service function domain. Overhead for, in this work developed and evaluated, partial encapsulation approaches (initial-partial and as required) is less than that of a full encapsulation (in all SF path segments). Overhead decrease (the difference between full and partial encapsulation in emulated topology) is up to 50 % **(38 bytes of overhead for a single packet versus 16 bytes as shown in Fig. 9 for NSH MD-Type 2 with context encapsulation)**. It conforms to all emulated encapsulation protocols – virtual local area network protocol (VLAN), multi-protocol label switching protocol (MPLS), and network service header protocol (NSH).

Section 3. Reactive SF path detection

Two ways of data packet classification are shown in Fig. 10. The proactive way is used in service function chaining architecture by default, but the reactive way is a developed and evaluated solution in this work.

A proactive way is when packet classification is performed only if a predefined manual static path policy for ingress network traffic is set by the network administrator. This way can be used if ingress traffic is previously known and expected.

The reactive way, developed in this work, allows for its modification at runtime. Here, ingress network traffic also has initial encapsulation, and it can be updated via subsequent classification, which is enacted by a service function (SF) if it cannot perform packet processing. The reactive way enables SF grouping in a hierarchy. Previously, service function chains could only be built from one of all available SF paths, but hierarchy allows grouping multiple SF paths to form a single chain.

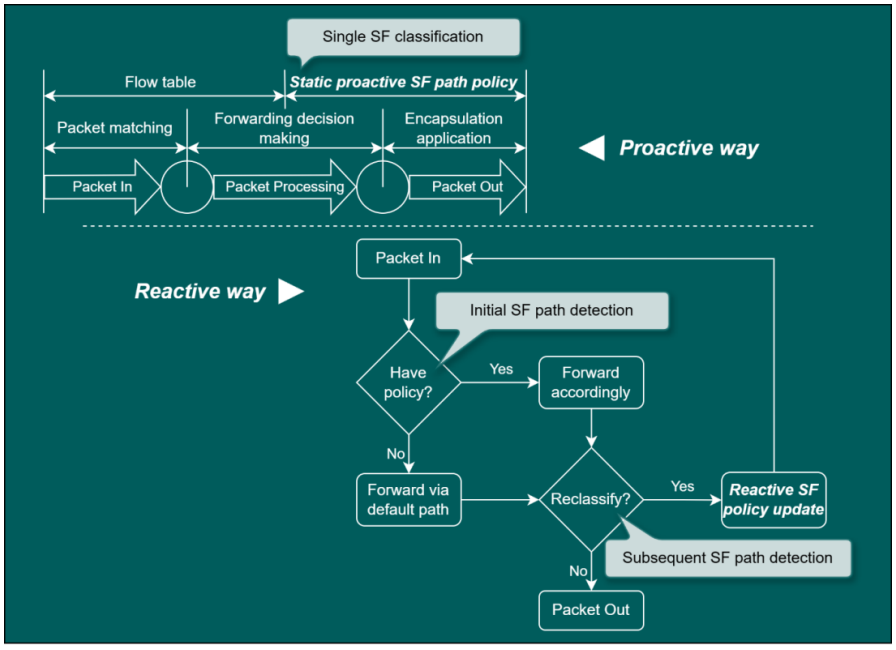

Fig. 10. Reactive SF path detection.

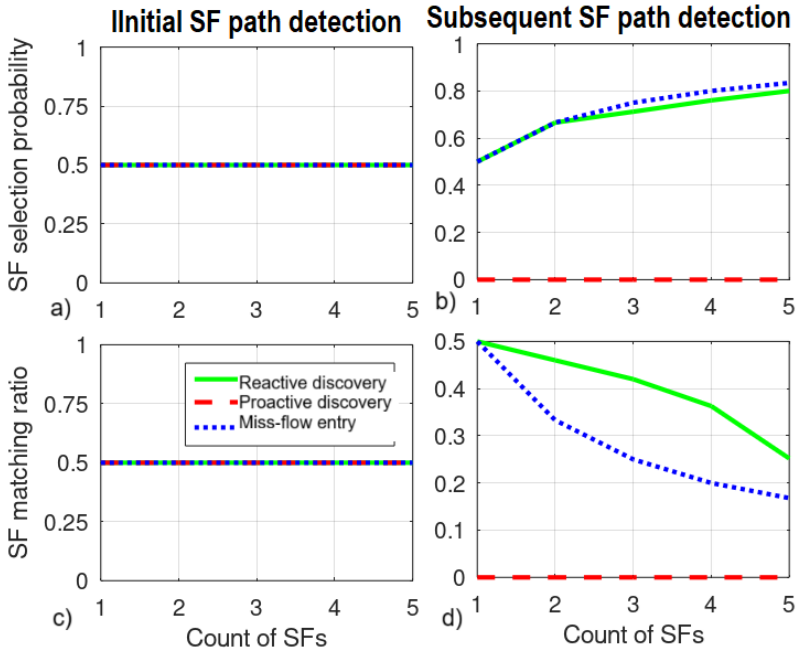

Fig. 11. Comparison of reactive and proactive ways:
a) probability to select the data flow appropriate SF in the initial classification;
b) probability to select a data flow appropriate SF in subsequent classification;
c) SF match discovery ratio in initial classification;
d) SF match discovery ratio in subsequent classification.

Figure 11 shows the difference in reactive and proactive SF path discovery ways. Graphs are directly dependent on the emulated network topology and characteristics of the service function domain. In the emulation environment, the SF path consisted of a single SF. A single-parameter classification was used. In other cases, the SF path can contain more SFs, and classification can be enacted again by other parameters than those parameters used in the initial classification. However, from emulation results, it is clear that the reactive way is capable of improving SF path discovery in the case of subsequent classification. Graphs a) and b) of Fig. 11 show the probability with which, as a result of classification, an appropriate SF path will be selected for ingress data flows. Graphs c) and d) show a ratio between all possible matchings and those that are appropriate matches. In accordance with characteristics, graphs a) and c) show a probability and matching ratio of 50 % equally for reactive and proactive ways, as there is only a single opportunity to make the classification (classification at the beginning of the SF chain). Whereas in graphs b) and d), a proactive way cannot do subsequent classification, and therefore, its value is 0. A reactive SF path discovery way, with the use of subsequent classification, can find an appropriate SF path. The probability of a reactive way in graph b) will never surpass the value 1, and its closing in on the maximal value can be explained by the fact that the possibility of not finding a valid SF path diminishes as the subsequent classification count increases. Unfortunately, in graph d), an inverted tendency can be observed as the valid SF path ratio against all possibilities shrinks with the increase of classification count.

Section 4. SF share in inter-data center networks

Not always are network resources exclusively used by one service provider. Even today, enterprises are placed in a multi-domain infrastructure. Virtual network function principle allows for customization of resource sharing and decreases management costs. Service providers can offer a set of infrastructure and applications as a platform on which enterprises can deploy their network applications. Within this platform, enterprises can develop their own service customizations that are suitable for their business needs [24].

Figure 12 shows the SF share in inter-data center networks. Communication is happening among three distant locations. Out of which two are multihomed (a state when a single target is reachable via multiple connection points at the same time), but the third is with a direct link to the second one through a leased line.

In this work developed and evaluated SF share with neighboring data center could optimize network device resource use (network equipment that is in idle state in one data center could serve a network traffic load which is directed to the other data center) and could enable a split allocation of systems (service function chain can be made of joining network functions of one data center with other network functions placed in a second data center). Peak loads and other unexpected network events can be managed without the reservation of additional resources for insurance.

Multipath TCP (MPTCP) is a version of the protocol that allows attaching multiple data flows to a single dialog, but steers them via different routes. MPTCP data flow, that belongs to a single dialog, is being divided into sub-flows and addressed to multiple IP addresses, but is steered through the same SF path at the receiving end, nullifying the benefits of division. SF share is a solution for this problem, too.

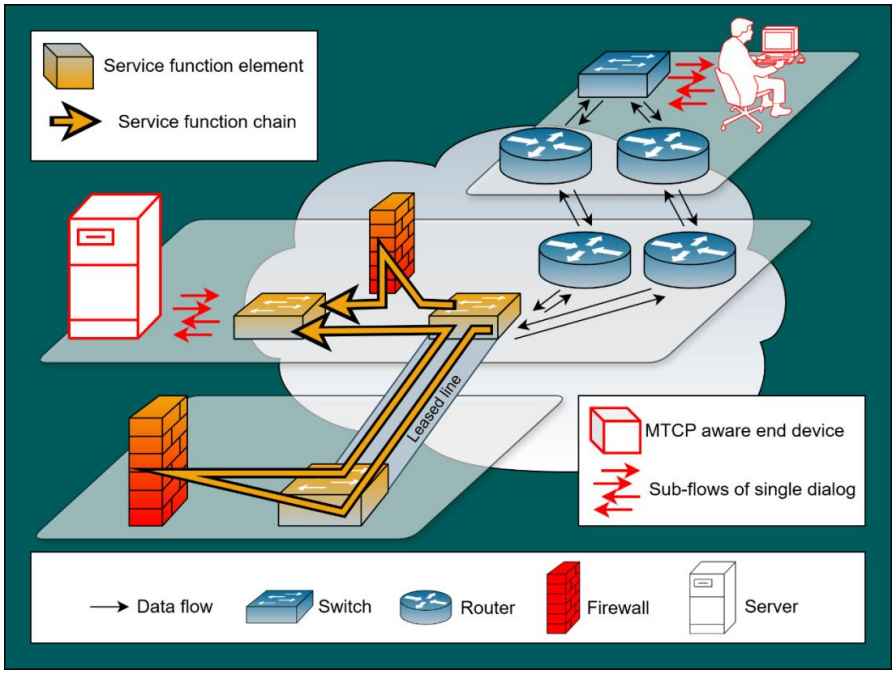

Fig. 12. SF share.

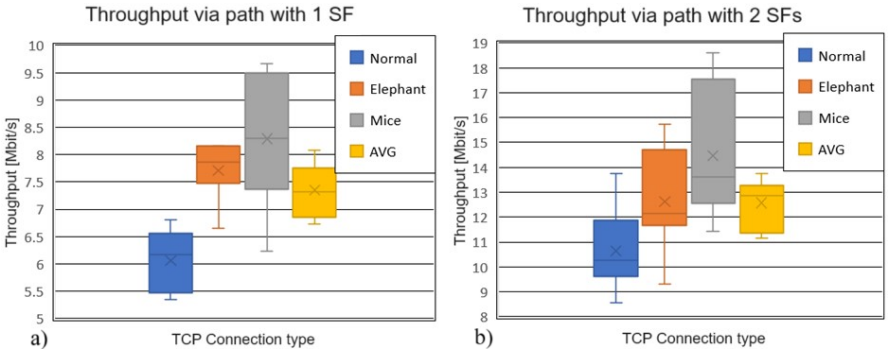

Fig. 13. Comparison of MPTCP flow throughputs:
a) multiple flows of a single dialog are steered via 1 SF;
b) multiple flows of a single dialog are steered via 2 SF in parallel.

Figure 13 shows network topology emulation results, where graph a) is the case when the whole dialog is steered to 1 SF, within a single data center, but graph b) shows the result of the developed and evaluated solution when flows of the dialog are steered to 2 SFs in parallel with the use of a connection between data centers (for example leased line). Results have a direct dependency on the topology, where in one case, 1 SF was used, but in the other case, 2 SF paths were used, and each contained 1 SF. In other cases, SF paths can contain multiple SFs, or more than two SF paths can be used. However, the results of the emulated topology clearly show an increase in throughput for all types of flows in the SF share solution. Increase (between the use of 1 SF and use of 2 SFs in parallel) is up to 60 % (**8 Mbit/s of throughput versus 14 Mbit/s for AVG (average of all flows)**). In both graphs, multiple types of flows are evaluated, which are categorized by flow length, having a period of 60 seconds for each measurement (normal – 1 dialog in 10 seconds, elephant – 1 dialog in 60 seconds, mice – 5 dialogs in parallel and each in 0.2 seconds, AVG – an average of all types). The biggest throughput is observed for mice flows.

Section 5. SR-MPLS on re-programmable data plane

A segment routing (SR) domain is a set of network nodes that participate in a source-based routing model. These nodes can be placed in a single physical infrastructure, or they can be allocated in remote locations. If multiple routing instances are in use, the SR domain usually accommodates all protocol instances in the network. Even though some applications tend to split the network into multiple SR domains, each of which contains one or more routing instances, it is supposed that all network nodes within an SR domain are managed by one and the same administrative element [25]. In this study, an SR-MPLS on P4 switches is developed and evaluated. For its realization, a software needed to be made, one that can be run on multiple network devices. A service function chaining (SFC) domain was implemented to emulate various segment routing paths in the topology. This allows for the evaluation of overhead made by the use of the MPLS protocol in segment routing.

In the emulated topology, P4 switches work as service function forwarders (SFFs) and hold the logic that emulates segment routing with MPLS (SR-MPLS) as transport for the creation of service function chaining (SFC). Emulated topology is shown in Fig. 14. Classifier (CL) attaches MPLS labels as SFC encapsulation for packets that are

received from the source (Src.) Service function proxy (SFP) removes the last label at the domain's egress, before the packet is transmitted to the destination (Dst). This research closes with a recommendation to describe the SFC with a single segment identifier (SID) as a single MPLS label, in opposition to the use of multiple labels.

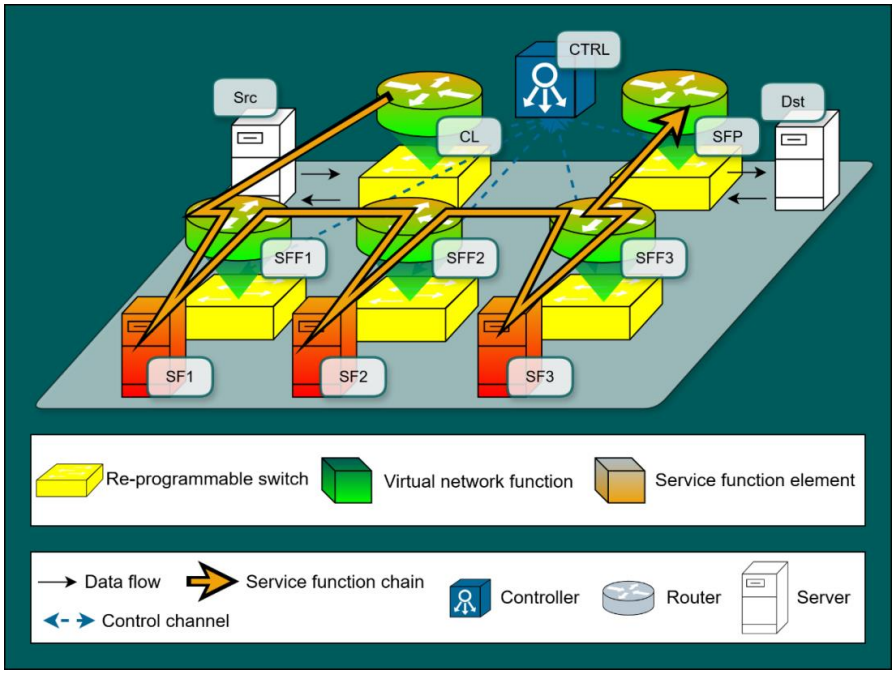

Fig. 14. SFC via SR-MPLS on P4.

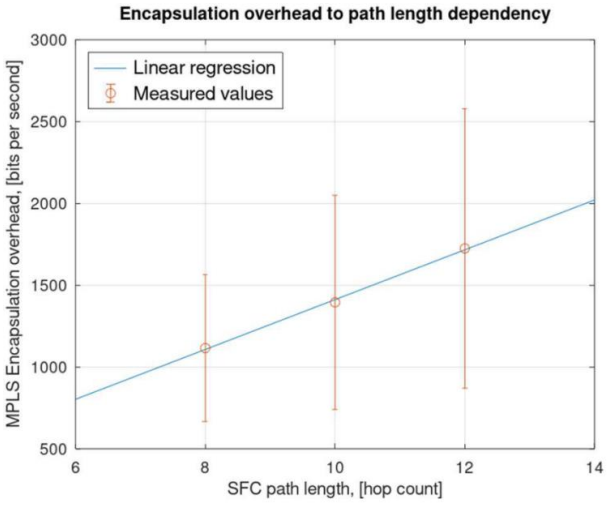

Fig. 15. Encapsulation overhead dependency on the path length.

MPLS encapsulation created an overhead dependency in bits per second, from the service function path hop count, as shown in Fig. 15. Results have a direct dependency on the emulated network topology (two-tier star topology with three service functions). In other cases, the topology type or service function count can vary. However, emulation results clearly show that overheads increase as the hop count gets bigger for longer SF paths.

The overhead in discussion refers to that which is created by MPLS encapsulation, which is attached to original data flows while they cross the service function domain. Evaluated increase (between an SF path of eight hops and an SF path of 12 hops) is approximately 40 % **(11000 bit/s versus 17000 bit/s of overhead in Fig. 15)**. The increase originates from two causes: an MPLS label stack is used for SF path description, and only an initial classification is used at the ingress of the SF domain. In this research, it is proposed to eliminate both mentioned causes by the use of a single segment identifier (MPLS label) to describe an SF path altogether.

In this study, a software for service function chaining via SR-MPLS on P4 switches was also developed and evaluated. Realized software is compatible with PISA architecture derivation *bmv1*, which is used in the mininet emulator. Software is available in a repository [10].

Section 6. Analysis of the Internet of Things

The Internet of Things (IoT) is a perpetually growing network of device mesh in which devices are directly or indirectly attached to the Internet. Each of the devices can be an autonomous solution or an integration into existing objects. The conception of IoT struggles with the definition of devices, thus allowing manufacturers to vary in device attribute incorporation. IoT mesh is a vertically fragmented system due to the complexity of the protocol stack, variation in data format, and multiple communication procedures [26].

In this study, IoT segmentation is proposed according to the environment or ecosystem they are attached to. Segmentation is shown in Fig. 16. Devices would be segmented into three groups as follows:

1. Standalone device – IoT functionality is the essence of a thing's existence.
2. Add-on device – IoT is an additional function to a device with a different main purpose.
3. Module – IoT functionality exists only if multiple devices cooperate among themselves.

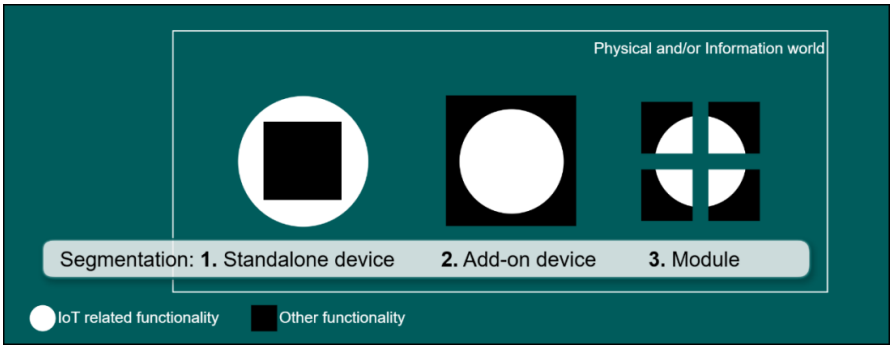

Fig. 16. IoT segmentation.

This research proposed a classification by physical size, shown in Fig. 17. The figure depicts size of devices relative to a grown human. The ratio could vary with the size of a child in case of fixed devices, but portable and wearable devices should be comfortable to carry for all people.

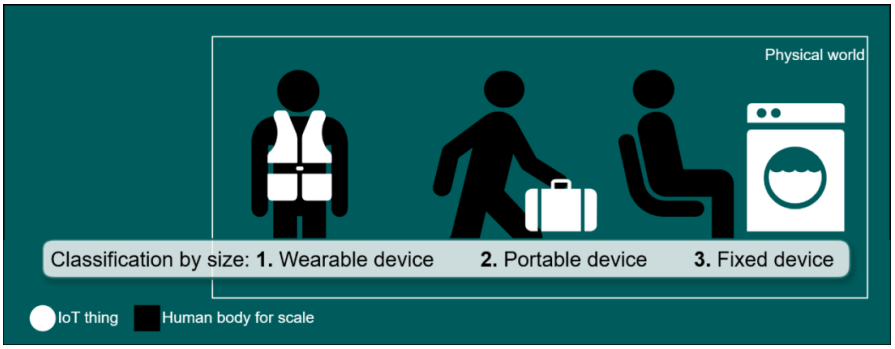

Fig. 17. IoT device classification by size.

In this work, 11 IoT network attributes are identified as follows:

- network devices are compatible;
- network is heterogeneous;
- network is erratic by nature;
- network is big in scale (IoT devices attached to public Internet, as described by other authors, is a problem due to their number);
- network device location must be traceable;
- network devices must be identifiable;
- network is self-healing (it needs to withstand the unreachability of some network nodes);
- in a network, the privacy of an individual is valued;
- network devices are easy to use with support of plug-and-play functionality;
- the network and its devices are manageable.

Attributes can be used partially, leading to network fragmentation and inability of the device to deliver interconnectivity.

Section 7. Thread mesh density considerations

In this work, the evaluation of Thread technology has been done via emulation of network topology. Thread mesh is observed while its density (connection count between devices) is changed to measure the effect that connection loss has on each of the topologies. Topologies are shown in Fig. 18. The goal of this research is to decrease regeneration frequency in personal area networks (PANs), which is caused by changes in Thread mesh connectivity. From the made emulations, the following scenario emerges – in the case of a Thread network disjoint, a stray end device initially searches for a routing-capable device that could attach it back to the PAN, but if the search is unsuccessful, it creates a new PAN for itself. When there is a chance for an initial set of network nodes to group, this cannot be done, as two PANs exist. For a return to a single PAN, a regeneration process must take place for all network devices. This means that even those devices that had not suffered from a connection loss must now terminate their stable connection to join the new PAN.

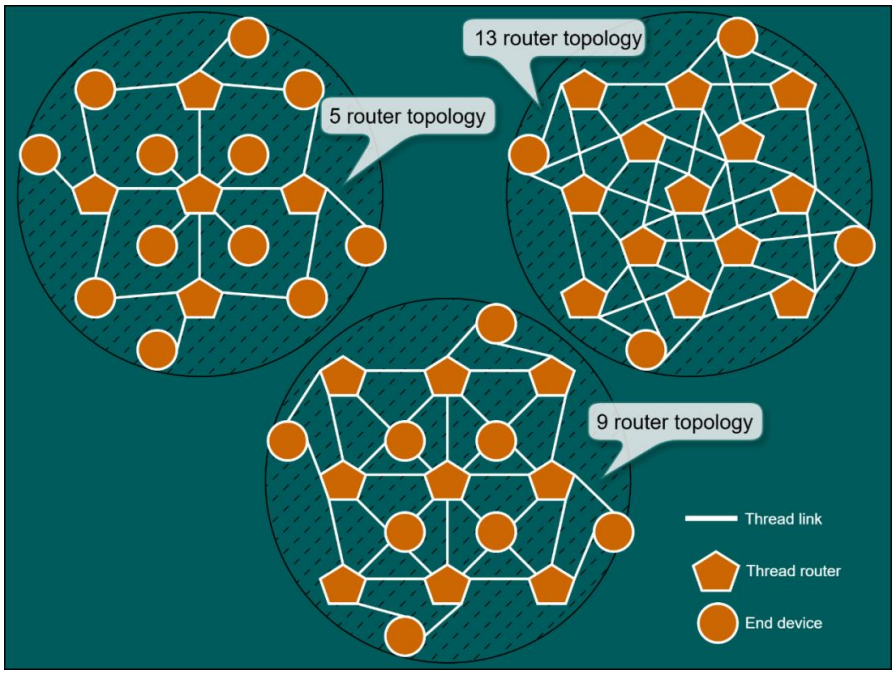

Fig. 18. Thread mesh density variations.

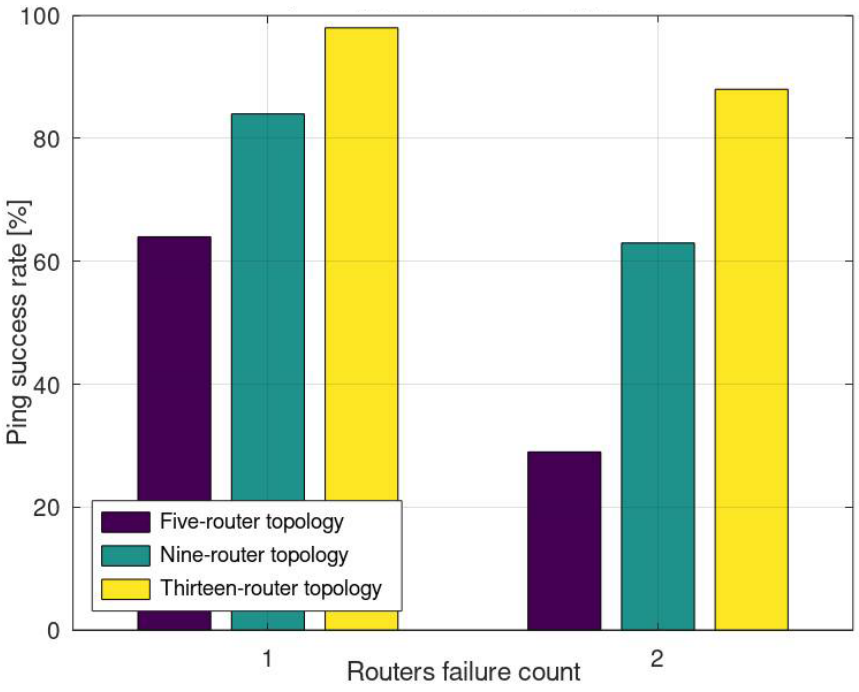

Fig. 19. Thread network reachability.

Measured emulated Thread network topologies' reachability is shown in Fig. 19. Even though only three topology types were emulated, due to the Thread mesh working principles, it is enough to obtain appropriate emulation results for other cases where the device count is bigger.

Reachability was measured by observation of the effect caused by the router disconnecting from the mesh. Even with two central routers disconnected, the 13-router topology is capable of maintaining connections between transmitting and receiving network nodes using backup paths. Experiments were done as follows:

- ping is sent from one end device to the other;
- take packet capture (capturing both directions of ping and Thread made communication messages);
- disconnect two central routers
- the runtime of the experiment is kept constant (10 min), in which disconnected routers are kept turned off.

The result of the experiment, in the case of a 13-router topology, is approximately 84 % successful ping data packet delivery.

The result of this research is a proposed recommendation – locate Thread network devices at a distance that allows for at least two backup connections to be formed for each end device. This density allows for dodging personal area network regenerations on link breakage (cases when routing-capable devices are disconnected from the network) as end devices have a sufficient count of backup paths.

Section 8. Intent-based networking analysis

It is stated that users' service requests need to be technology independent. That means the users do not know about the technologies that service forwarders are using. Therefore, users cannot make a request for the use of a specific technology, but are provided with the ability to only modify the service that is provided [27].

Intent-based networking (IBN) supports such a working principle. Its main goal is to use artificial intelligence to tackle cumbersome network questions (configuration creation, maintenance of connections between network nodes, security assurance, etc.).

To advance intent-based networking incorporation in network control management, this research discovers a necessity for change in the model's concept. The performed research identifies the existing model's structure and proposes the creation of two new derivatives that are shown in Fig. 20.

Figure 20 shows the existing default IBN model a), which does not have a defined network management type. Within this research, the following model's derivatives are proposed:

1. IBN model b) for autonomous networks – for use in autonomous networks where network creation and maintenance processes do not require human oversight in network management. These networks are oriented on connectivity assurance and efficiency monitoring. A change in the network is made only if it is necessary to enhance transmission parameters or to add or remove network nodes.
2. IBN model c) for supervised networks – for use in networks whose behavior is defined and modified in accordance with the administrator's requests. These networks are paid services and enhanced security-oriented and are modified depending on audit results about processes that are (or may be) decoupled from network performance efficiency.

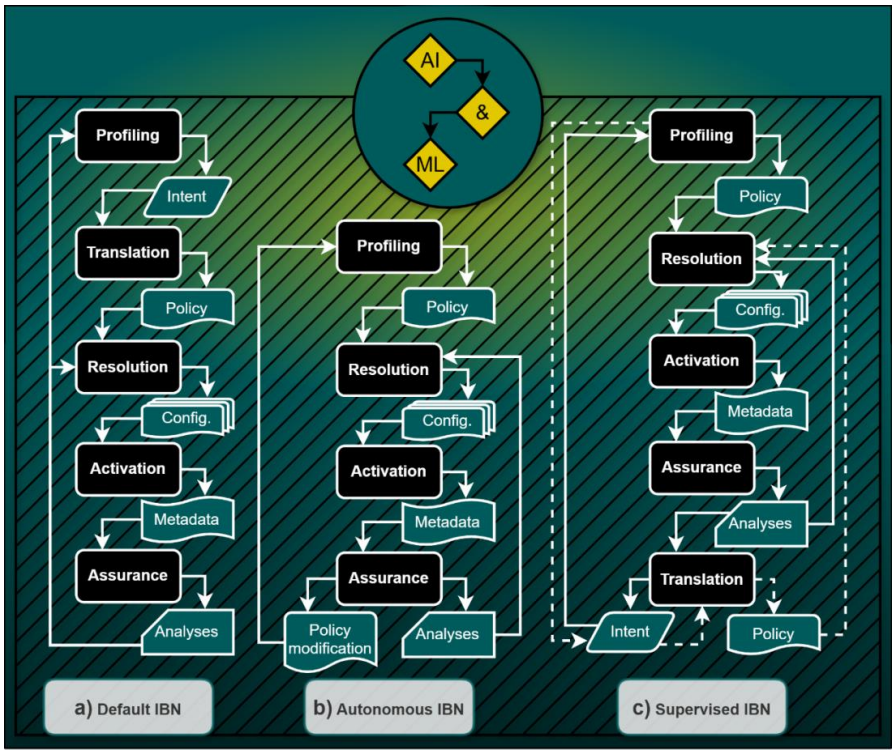

Fig. 20. IBN models:
a) default IBN model;
b) IBN model for autonomous networks;
c) IBN model for supervised networks.

In this research, already existing IBN applications are analyzed and their proposed standardization drafts are discussed. Most importantly, IBN structure elements (profiling block, translation block, resolution block, activation block, assurance block) and their interconnectivity are revealed. From these discoveries, it is concluded that overly cumbersome network management task halts IBN adaptation.

CONCLUSIONS

In this Thesis, eight related studies are performed. They all are done in packet-switched networks for an enhancement of the following telecommunication technologies:

- software-defined networking (SDN);
- network function virtualization (NFV);
- service function chaining (SFC);
- programmable protocol-independent packet processing (P4);
- intent-based networking (IBN);
- the Internet of Things (IoT).

From the conducted research results, multiple important aspects are concluded about the working principle of studied technologies, which point to shortcomings in them, possible optimization, and ways for the use of new solutions to mitigate the shortcomings.

It has been concluded that by default, the service function chaining (SFC) domain holds multiple shortcomings – it does not define conditions for encapsulation application, it functions only thanks to a predefined manual service function (SF) path policy, and it has no definition of functionality for multihomed tenancy. For the mitigation of stated shortcomings, solutions are proposed – network topology-aware SFC encapsulation, reactive SF path policy creation, and SF share in inter-data center networks.

It has been found that, by default, the Internet of Things defines 11 various attributes (reachability, security, identification, continuity, etc.), without any connection to physical device parameters (dimensions, positioning, data transmission type, etc.), thus fragmenting interconnectivity among devices. Partially, this problem is mitigated by Thread and Matter technologies that enable centralization of device management in a small local area network, but Thread itself suffers from link breakages that are started by its network regeneration process. Solutions proposed are – a recommendation for IoT device categorization, and a recommendation for Thread network density.

It has been concluded that programmable switch architecture (PSA) enables virtual realization of network functions, but program code needs to be developed for the definition of desirable functionality. By creation of SFC functionality on top of PSA, it has also been concluded that SF paths made by use of segment routing with MPLS (SR-MPLS) have dependency on encapsulation label count. A program code is developed for SFC creation on PSA via SR-MPLS encapsulation [10], and after evaluation of the experimental results, a recommendation is proposed for SF path declaration using a single segment identifier.

It has been concluded that the intent-based networking (IBN) model consists of multiple building blocks – profiling, translation, resolution, activation, and assurance – each of which is realized with the use of artificial intelligence. An assurance block that is committed to the provision of a network state forecast cannot be realized, as existing artificial intelligence functionality is incapable of accomplishing the given task. To ease the prediction generation task given to , two alternative IBN models are proposed.

The research objective (perform an analysis of existing solutions and propose new adaptive solutions for telecommunications technologies) and its related tasks result in

the development and evaluation of seven adaptive solutions for four telecommunication problems.

Achieved results are listed in relation t to hypotheses:

1. **Reactive SF path discovery and coordination of transmitted information along multiple administrative domains is achievable via the use of service function chaining.** Hypothesis derives from the problem of enhancing network path generation and coordination of transmitted information in them using service function chaining. The proposed solutions are:

1.1. A service function path encapsulation according to the necessity for service function chaining.

Two data packet encapsulation methods are developed in this work. They allow for imposing an encapsulation only in service chain segments where it is necessary, opposite to encapsulation in all segments of the chain. It allows to decrease overhead (a 50 % overhead decrease is achieved in emulated network topology that is 38 bytes of overhead for single packet versus 16 bytes, as shown in Fig. 9, for NSH MD-Type 2 with context encapsulation), enables multihomed tenancy, and removes service path limitations (segment count dependency from header, SF path steering only within a single administrative domain).

(Solution results from the research are presented in Appendix 1. The explanation is given in Section 2. Background technologies are introduced in Subsections 1.2 and 1.3.)

1.2. A reactive service path discovery using a default path and data flow reclassification.

Solutions developed in this research are a combination of using the default path and network data packet re-classification, in opposition to predefined policy. This mitigates human intervention in service function path creation. **The author's proposed solution is capable of subsequent SF path discovery for single-conditioned policy-steered network traffic, while the existing solution is not.**

(Solution results from the research are presented in Appendices 2 and 3. Its explanation is given in Section 3. Background technologies are introduced in Subsections 1.1 and 1.3.)

1.3. Service function share in inter-datacenter networks.

Solution developed in this research allows steering a single multipath TCP (MPTCP) belonging to sub-flows via different service functions (SFs), in opposition to single SF usage. The SF share can be put in use in inter-data center networks to increase overall connection throughput (in an emulated network topology, a 60 % throughput increase was measured, that is 8 Mbit/s of throughput versus 14 Mbit/s for AVG (average of all flows) value in Fig. 13).

(Solution results from the research are presented in Appendix 6. Its explanation is given in Section 4. Background technologies are introduced in Subsections 1.1 and 1.3.)

2. **Fragmentation of the Internet of Things can be mitigated by using a unified transmission medium, and device compatibility can rise with the use of a unified application layer, if more specific characteristics of things are defined.** Hypothesis derives from the problem of fragmentation mitigation of

the Internet of Things by using unified communication standards for the increase of compatibility among devices in use. The proposed solutions are:

2.1. Recommendation on the implementation of specific characteristics for IoT device definition.

IoT mesh is a vertically fragmented system due complexity of the protocol stack, variation in data format, and multiple communication procedures [26]. Therefore, in this research recommendations are developed for categorization of IoT devices according to their physical size **(wearable, portable, fixed)**, and according to IoT functionality role in relation to the device's main purpose fulfillment **(standalone, add-on, module)**, and according to communication parameters **(what physical signal it can transmit)**. The author's proposed classification allows overcoming the fragmentation of the network relating to a lack of interconnectivity among devices.

(Solution results from the research are presented in Appendix 4. Its explanation is given in Section 6. Background technology is introduced in Subsection 1.4.)

2.2. Recommendation on IoT Thread mesh network density compliance.

Developed in this work is a recommendation stating that it is necessary to locate Thread network devices at a distance that allows for at least two backup paths to be created. This density decreases personal area network regeneration frequency on connection loss (in the emulated network topology selected density provided the connection continuity of 84 %, which equals over 500 pings received in a period of 10 min at two central routers disconnected by measuring packet loss). **There is no set recommendation for link count besides that which is proposed in this research.**

(Solution results from the research are presented in Appendix 5. Its explanation is given in Section 7. Background technology is introduced in Subsection 1.5.)

3. **Service function chaining via segment routing can be built on top of programmable devices, thus serving as an alternative to the use of purpose-specific or proprietary devices with similar capabilities.** Hypothesis derives from the problem of the cut on the use of task-specific or proprietary devices by the advancement of the feature set for re-programmable units. The proposed solution is – **service function path declaration with a designated segment identifier in segment routing. Also, the author's built software can be reused as an alternative to proprietary solutions with similar capabilities.**

   In this work, a software for SFC via SR-MPLS on P4 switches was developed and evaluated and is available in a repository [10]. It can be concluded that the SF path declaration needs to be made by using a single MPLS label for all SF paths, if transport is segment routing. This allows for dodging the increasing overhead for longer SF paths (in emulation, a 40 % increase is observed when the SF path with eight hops and the SF path with 12 hops are compared, that is 11 000 bit/s versus 17 000 bit/s of overhead in Fig. 15).

   (Solution results from the research are presented in Appendix 8. Its explanation is given in Section 5. Background technologies are introduced in Subsections 1.6 and 1.7.)

4. **Computer network creation and management can be automated by using intent-based networking if artificial intelligence functionality is delegated separately to different network management types.** Hypothesis derives from the problem of automation of human-managed network configuration, maintenance, and telemetry analysis by the provision of intent-based

networking. The proposed solution is – **recommendation for facilitating intent-based networking adaptation.**

In this work, a split of the IBN model into two sub-models is proposed. One for autonomous networks and the other for supervised networks. The division would ease the tasks of assurance block (an action set that the use of artificial intelligence provides analysis and forecast of network performance). **Studied references of Appendix 7 (works of other authors) strongly suggest the need for ease of tasks of the assurance block. The proposed solution is yet to be tested.**

(Solution results from the research are presented in Appendix 7. Its explanation is given in Section 8. Background technology is introduced in Subsection 1.8.)

Developed and evaluated adaptive solutions are the result of multiple completed studies. Hypotheses defined in this Thesis have been found to be true by conducting analysis and experimental evaluation by emulation of proposed solutions application for given problems in various telecommunication technologies. This Thesis and its results together can be used as a forging ground for future research.

REFERENCES

Listed references are compliant with usage in the Thesis summary.

[1] United Nations General Assembly, Transforming our world: the 2030 Agenda for Sustainable Development, A/RES/70/1, October 21, 2015. [Online]. Available: https://sustainabledevelopment.un.org/post2015/transformingourworld, last viewed May 15, 2025

[2] National Physical Laboratory of the United Kingdom, "Packet Switching: The First Steps on the Road to the Information Society." [Online]. Available: https://www.npl.co.uk/getattachment/about-us/History/Famous-faces/Donald-Davies/UK-role-in-Packet-Switching-(1).pdf.aspx, last viewed May 15, 2025

[3] Mihaeljans, M., Skrastiņš, A. Network Topology-aware Service Function Chaining in Software Defined Network. In: 2020 28th Telecommunications Forum (TELFOR 2020): Proceedings, Serbia, Belgrade, 24–25 November 2020. Piscataway: IEEE, 2020, pp. 1–4. ISBN 978-1-6654-0500-3. e-ISBN 978-1-6654-0499-0. Available from: doi: 10.1109/TELFOR51502.2020.9306554

[4] Mihaeljans, M., Skrastiņš, A. Reactive Service Function Path Discovery Approach in Software Defined Network. In: 2021 29th Telecommunications Forum (TELFOR 2021): Proceedings, Serbia, Belgrade, 24–24 November 2021. Piscataway: IEEE, 2021, pp.29-32. ISBN 978-1-6654-2586-5. e-ISBN 978-1-6654-2585-8. Available from: doi: 10.1109/TELFOR52709.2021.9653356

[5] Mihaeljans, M., Skrastiņš, A. Evaluation of Reactive Service Function Path Discovery in Symmetrical Environment. Telfor Journal, 2022, Vol. 14, No. 1, pp. 2–7. ISSN 1821-3251. e-ISSN 2334-9905. Available from: doi: 10.5937/telfor2201002M

[6] Mihaeljans, M., Skrastiņš, A. IoT Concept and SDN Fusion in Consumer Products: Overview. In: 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2023), Spain, Tenerife, 19–21 July 2023. Piscataway: IEEE, 2023, pp. 1–6. ISBN 979-8-3503-2298-9. e-ISBN 979-8-3503-2297-2. Available from: doi: 10.1109/ICECCME57830.2023.10252518

[7] Mihaeljans, M., Skrastiņš, A. Efficient Multipath Service Function Chaining in Inter-Data Center Networks. In: 2023 31st Telecommunications Forum (TELFOR 2023): Proceedings, Serbia, Belgrade, 21–22 November 2023. Piscataway: IEEE, 2023, pp. 1–4. ISBN 979-8-3503-0312-4. e-ISBN 979-8-3503-0313-1. Available from: doi: 10.1109/TELFOR59449.2023.10372615

[8] Mihaeljans, M., Skrastiņš, A. Openthread Network Density Evaluation: Quantitative Analysis. In: 2023 Symposium on Internet of Things (SIoT 2023): Proceedings, Brazil, São Paulo, 25-27 October, 2023. Piscataway: IEEE, 2023, pp. 1–5. ISBN 979-8-3503-2944-5. e-ISBN 979-8-3503-2943-8. Available from: doi: 10.1109/SIoT60039.2023.10390236

[9] Mihaeljans, M., Skrastiņš, A., Poriņš, J. Service Function Chaining via SR-MPLS over P4: A Rudimentary Analysis. In: 2024 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS 2024): Proceedings, Croatia, Dubrovnik, 24–27 September 2024. Piscataway: IEEE, 2024, pp. 1–8. ISBN 979-8-3503-5470-6. e-ISBN 979-8-3503-5469-0. Available from: doi: 10.1109/ICCNS62192.2024.10776149

[10] Mihaeljans, M., Skrastins, A. P4 program code used for SFC via SR-MPLS on PSA, [Online]. Available: https://github.com/MartinsMihaeljans/SFC-on-P4/tree/main, last viewed June 2024

[11] Mihaeljans, M., Skrastiņš, A., Poriņš, J. Paramounts of Intent-Based Networking: Overview. Elektronika ir elektrotechnika/Electronics and Electrical Engineering, 2024, Vol. 30, No. 6, pp. 53–59. ISSN 1392-1215. e-ISSN 2029-5731. Available from: doi: 10.5755/j02.eie.38680
[12] Boutaba, R., Aib, I. "Policy-Based Management: A Historical Perspective," Journal of Network and Systems Management, vol. 15, no. 4, pp. 447–480, Dec. 2007.
[13] Connectivity Standards Alliance, Matter Overview, p. 2, 2024.
[14] Connectivity Standards Alliance, Matter Specification R1.3, Document 23-27349, p. 49, Apr. 17, 2024.
[15] Liu, Z., Cui, P., Hu, Y., Dong, Y., Tang, K., and Xue, L. "A programmable data plane that supports definable computing," 2021 International Conference on Advanced Computing and Endogenous Security, Nanjing, China, 2022.
[16] Ali, Z., Filsfils, C., Camarillo, P., Voyer, D., Matsushima, S., Rokui, R., Dhamija, A., and Maheshwari P. Building Blocks for Network Slice Realization in Segment Routing Network, Internet-Draft, TEAS Working Group, p. 4, Sep. 7, 2022.
[17] Clemm, A., Ciavaglia, L., and Granville, L. Z. "Intent-Based Networking – Concepts and Definitions," IRTF RFC 9315, 2022, Available from: doi: 10.17487/RFC9315
[18] Li, C. et al., " Intent Classification draft-li-nmrg-intent-classification-00," IETF Network Working Group, 2019.
[19] Hares, A. "Intent-Based Nemo Overview draft-hares-ibnemo-overview-01," IETF Internet-Draft, 2016.
[20] Leivadeas, A., and Falkner, M. "A Survey on Intent-Based Networking," in IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 625–655, Firstquarter 2023, doi: 10.1109/COMST.2022.3215919.
[21] Leivadeas, A., and Falkner, M. "Autonomous Network Assurance in Intent Based Networking: Vision and Challenges," 2023 32nd International Conference on Computer Communications and Networks (ICCCN), Honolulu, HI, USA, 2023, pp. 1–10, Available from: doi: 10.1109/ICCCN58024.2023.10230112.
[22] Stevens, N., Weiss, W., Mahon, H., Moore, B., Strassner, J., Waters, G., Westerinen, A., and Wheeler, J. Policy Framework, Internet-Draft, Expires Mar. 2000, p. 18, Sep. 13, 1999.
[23] ETSI, Network Functions Virtualisation – Introductory White Paper, Issue 1, p. 11, 2013.
[24] ETSI, Network Functions Virtualisation (NFV); Use Cases, ETSI GS NFV 001 V1.1.1, p. 21, Oct. 2013.
[25] Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and Shakir, R. Segment Routing Architecture, RFC 8402, p. 4, Jul. 2018.
[26] Rafique, W., Qi, L., Yaqoob I., and Imran, M., Rasool, R., Dou, W. “Complementing IoT Services Through Software Defined Networking and Edge Computing: A Comprehensive Survey” in Communications Surveys & Tutorials, Vol. 22, No. 3, 2020.
[27] Wu, Q., Liu, W., and Farrel, A. Service Models Explained, RFC 8309, p. 9, Internet Engineering Task Force (IETF), Jan. 2018.

Mārtiņš Mihaeljans was born in 1994. He received a Bachelor's degree in Electrical Science (2018) and a Master's degree in Telecommunications (2020) from Riga Technical University. His research interests are related to packet-switched computer networks.